

SYSTEM, METHOD, AND PROGRAM FOR ESTABLISHING MODEM
COMMUNICATION BETWEEN A MASTER COMPUTER SYSTEM AND A
PLURALITY OF SLAVE COMPUTER SYSTEMS THROUGH A COMMON
SERIAL COMMUNICATION CHANNEL

5

BACKGROUND OF THE INVENTION

1. Technical Field:

10 The present invention relates in general to data processing systems, and, in particular, to a system, method, and program for establishing communication among data processing systems. Still more particularly, the present invention relates to a system, method, and program for establishing modem communication between a master computer system and a plurality of slave computer systems through a common serial communication channel.

2. Description of the Related Art:

A data processing system typically includes a system processor, a memory system, and an input/output ("I/O") controller. The data processing system may be coupled in communication with another data processing system. Two data processing systems communicate by coupling a communication device, such as a modem, to the I/O controller of each data processing system and linking the communication devices or modems together through a serial communication channel, such as a telephone line or a wireless communication channel. Interconnect among these devices is usually accomplished using a public switch telephone network.

In one application, the two data processing systems

are coupled in communication with each other so that one data processing system directs and executes commands to the other data processing system. For example, one data processing system may be a master computer system that includes at least a master system processor, a master memory system, and a master I/O controller. The other data processing system may be a slave computer system which includes at least a slave system processor, a service processor, a slave memory system, and a slave I/O controller. The service processor is a dedicated special-purpose processor that serves as a full time hardware and software system monitor during the operation of the slave computer system. The service processor monitors the environmental status of the slave computer system and executes various operational or maintenance routines, such as re-configuration of the slave system processor or the slave computer system, execution of the slave computer system under a different configuration, or analysis of performance of the slave computer system and components or sub-systems of the slave computer system. The master computer system is coupled in communication with the slave computer system so that the master system processor is able to direct commands from the master computer system to execute various routines by the service processor at the slave computer system.

When the data processing system is coupled in communication with a plurality of other data processing systems, a separate serial communication channel (e.g., phone line or wireless channel) exists between the data processing system and each of the other data processing systems. However, a plurality of serial communication channels for communicating between the data processing system and each of the various other data processing systems means that the use of a larger amount of

communication resources is required. The communication resources are expensive and are not generally used on a regular basis.

5 In order to reduce expenses and a large amount of communication resources, a single communication channel can be used. The single communication channel couples the data processing system to a centralized hardware box, and the various other data processing systems are also able to couple in communication with the centralized hardware box through a communication switch. The centralized hardware box manages the communication between the data processing system and one of the other data processing systems.

10 Normally, this communication management is done manually requiring an operator to be present. If it is done electronically, then the data processing system initiates a session request or call for a particular data processing system to which the data processing system requests to communicate by sending an indication, such as a certain type of telephonic ring or distinguishing signal, to the centralized hardware box through the single communication channel. The centralized hardware box controls the switch to activate the communication between the data processing system and the requested particular data processing system by switching on the communication channel between the centralized hardware box and the particular data processing system. The existence of the centralized hardware box, however, results in the use of additional hardware and adds to the expense for providing communication between the data processing systems. The present invention recognizes the need and desire to eliminate hardware and reduce expenses associated with providing communication and manual intervention between data processing systems.

SUMMARY OF THE INVENTION

A system, method, and program for establishing modem communication between a master computer system and a plurality of slave computer systems coupled to a common serial communication channel are disclosed. An unique identification number is assigned to each of the slave computer systems. Modems of the slave computer systems are initialized to a "receive mode." The master computer system directs a session request to the slave computer systems through the common serial communication channel. The plurality of slave computer systems receive and respond to the session request by changing each of the modems to the "answer mode."

After the slave computer systems respond, the master computer system requests communication with a particular slave computer system among the plurality of slave computer systems. The master computer system uses the unique identification number to establish communication with the particular slave computer system by sending a request from the master computer system specifying the unique identification number of the particular slave computer system. After the master computer system requests communication with the particular slave computer system, communication is maintained between the master computer system and only the particular slave computer system by maintaining the modem for the particular slave computer system in the "answer mode" and switching the modems for all other slave computer systems to the "receive mode." After communication is established between the master computer system and the particular slave computer system, the master computer system controls the particular slave computer system by issuing commands to obtain the information.

The above as well as additional objects, features, and advantages of the present invention will become apparent in the following detailed written description.

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figure 1 is an exemplary block diagram of hardware for an overall system that includes a master computer system that establishes communication with a particular one of a plurality of slave computer systems in accordance with the present invention;

Figure 2 is an exemplary block diagram of a data processing system, which may be the master computer system or one of the slave computer systems in **Figure 1**, that is used to implement the present invention; and

Figure 3 is a flow chart of an exemplary method and program executed by the overall system of **Figure 1** to establish communication between the master computer system and a particular slave computer system in accordance with the present invention.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENT

The present invention discloses a system, method, and program for establishing communication between data processing systems. The present invention also discloses a system, method, and program for establishing modem communication between a master computer system and a plurality of slave computer systems coupled to a common serial communication channel. The present invention is not in any way limited to data processing systems or computer systems that are master computer systems or slave computer systems, but the present invention may be implemented with all suitable data processing systems or computer systems.

With reference now to **Figure 1**, an exemplary block hardware diagram of an overall system 98 that includes a master computer system ("master system") 100 establishing communication with a particular one among a plurality of slave computer systems ("slave servers") 1, 2, and 3 through a common communication channel 150 in accordance with the present invention is shown. Master system 100 is coupled to common communication channel 150, such as a serial communication channel. Exemplary communication channels include a telephone line for telephonic communication and a wireless channel for wireless communication. Common communication channel 150 is further coupled to slave servers 1, 2, and 3.

Master system 100 includes at least a master system processor 102, a master memory system 104, an input/output controller ("I/O contr.") 106, and a master modem 108. Master system processor 102, master memory system 104, and

I/O controller 106 are attached to bus 109 to enable communication therebetween, and master modem 108 is coupled to I/O controller 106. Common communication channel 150 is coupled to master modem 108.

5

Slave server 1 has at least a slave system processor 1A, a modem 1B, an I/O controller 1C, a slave service processor 1D, and a slave memory system 1E. Slave system processor 1A, slave service processor 1D, I/O controller 1C, and slave memory system 1E are coupled to a bus 1F to provide bus communication between these components. Slave server 1 is coupled to master system 100 through modem 1B and common communication channel 150. As shown in **Figure 1**, slave servers 2 and 3 are similarly configured.

10

Referring now to **Figure 2**, an exemplary block diagram of a data processing system 210, which may be master system 100 or slave computer system 1, 2, or 3 in **Figure 1**, is shown. The illustrative embodiment depicted in **Figure 2** is a workstation or server computer system such as the RS/6000® manufactured by International Business Machines Corporation (IBM) of Armonk, New York; however, as will become apparent from the following description, the present invention is also applicable to other data processing systems.

20

25

As illustrated in **Figure 2**, data processing system 210 (e.g., master system 100 or slave server 1, 2, or 3) includes at least one system processor 212 (e.g., master system processor 102 or slave system processor 1A, 2A, or 3A), which is coupled to processor bus 214. System

30

processor 212, which may comprise one of the PowerPC™ line of processors produced by IBM an Intel processor, etc., is a general-purpose processor that processes data under the control of an operating system and application software stored in memory system 216 (e.g., master memory system 104). System processor 212 is coupled via processor bus 214 and host Peripheral Component Interconnect (PCI) bridge 218 to PCI local bus 220.

Communication on PCI local bus 220 is governed by local PCI controller 227, which is in turn coupled to a non-volatile random access memory (NVRAM) 213 via a memory bus 215. Local PCI controller 227 is further coupled via a second host bridge 221 and a service processor bus 224 to a service processor 211 (e.g., slave service processor 1D, 2D, and 3D). Service processor 211 is a dedicated special-purpose processor that serves as a full-time hardware and software system monitor during the operation of data processing system 210. The functionality of service processor 211 is governed by system monitoring software stored in a service processor dynamic random access memory (SP DRAM) 222. At startup, service processor 211 boots from code in a flash memory 223. Flash memory 223 also stores system firmware that is executed by system processor 212 at startup prior to the loading of the operating system.

Data processing system 210 further includes an Industry Standard Architecture (ISA) bus 218, which is coupled to PCI local bus 220 by ISA bridge 217. Coupled to ISA bus 218 is an input/output (I/O) controller 219

(e.g., I/O controller 106, 1C, 2C, or 3C), which controls communication between data processing system 210 and attached peripheral devices such as a keyboard or mouse via a keyboard/mouse interface 240 and a disk drive through a disk drive interface 242. In addition, I/O controller 219 supports external communication by data processing system 210 via a serial port 236 and a parallel port 238. A modem, such as modem 108, 1B, 2B, or 3B, can also be attached to I/O controller 219.

10

I/O controller 219 includes a control logic 230, a dedicated system universal asynchronous receiver transmitter (UART) 232, and a transmitter ("T") 234. U.S. Patent Application No. 09/046,837 entitled (IBM Docket No. AT9-98-010) "METHOD AND SYSTEM FOR SHARING A SINGLE UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART) AMONG MULTIPLE RESOURCES IN A DATA PROCESSING SYSTEM" filed on March 24, 1998, which is assigned to International Business Machines ("IBM") of Armonk, New York, discloses such an exemplary I/O controller. UART 232 converts data received from control logic 230 from parallel format into serial format for communication via serial port 236 and converts data received from serial port 236 from serial format into parallel format prior to passing the received data to control logic 230. Transceiver ("T") 234 adjusts the voltage levels of in-coming and out-going signals to comply with the voltage specifications of data processing system 210 and the selected serial communication protocol (e.g., RS-232). Of course, depending upon design considerations, I/O controller 219 may include additional serial ports, each having a single associated UART and transceiver. Transceiver 234 has an enable input coupled

25

30

40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210 215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 295 300 305 310 315 320 325 330 335 340 345 350 355 360 365 370 375 380 385 390 395 400 405 410 415 420 425 430 435 440 445 450 455 460 465 470 475 480 485 490 495 500 505 510 515 520 525 530 535 540 545 550 555 560 565 570 575 580 585 590 595 600 605 610 615 620 625 630 635 640 645 650 655 660 665 670 675 680 685 690 695 700 705 710 715 720 725 730 735 740 745 750 755 760 765 770 775 780 785 790 795 800 805 810 815 820 825 830 835 840 845 850 855 860 865 870 875 880 885 890 895 900 905 910 915 920 925 930 935 940 945 950 955 960 965 970 975 980 985 990 995 1000

5

to service processor 211. By asserting and deasserting the enable inputs of transceiver 234, service processor 211 determines which of service processor 211 and system processor 212 is able to transmit and receive data via serial port 236. I/O controller 219 may just as easily be attached to the service processor bus 224.

10

With reference now to **Figure 3**, a flow chart of an exemplary method 300 and program executed by overall system 98 of **Figure 1** for establishing communication between master system 100 and a particular one slave server among slave servers 1, 2, and 3 in accordance with the present invention is shown. Exemplary method 300 discloses an example in which communication is established between master system 100 and slave server 2. However, exemplary method 300 is not in any way limited to establishing communication between master system 100 and slave server 2, and exemplary method 300 may be implemented and executed to establish communication between any suitable data processing system and another suitable data processing system.

25

30

Exemplary method 300 starts at block 302 and then proceeds to block 304, which shows the assignment unique identification numbers (UINs) 11, 22, and 33 to respective slave servers 1, 2, and 3. UINs 11, 22, and 33 are stored as UIN bit values in respective slave memory systems 1A, 2A, and 3A, which may be non-volatile random access memories ("NVRAMs"), for slave servers 1, 2, and 3. Method 300 then moves to block 306, which represents master system 100 storing UINs 11, 22, and 33 in

association with the identities of the associated slave, for example, servers 1, 2, and 3 in a table within master memory system 104 of master system 100. Following block 306, block 308 depicts modems 1B, 2B, and 3B of respective slave servers 1, 2, and 3 are each initially set to a "receive" mode (e.g., "on-hook" mode) while awaiting receipt of a session request.

Method 300 next moves to block 310, which illustrates master system 100 directing a session request to the plurality of slave servers 1, 2, and 3 through I/O controller 106, master modem 108, and common communication channel 150. Following block 310, block 312 shows modems 1B, 2B, and 3B of respective slave servers 1, 2, and 3 receive and respond to the session request by each changing from the receive mode to an "answer" mode (e.g., "off-hook" mode) that responds to the session request. Method 300 then proceeds to block 314, which represents that, after slave servers 1, 2, and 3 respond to the session request, master system 100 requests to establish communication with slave server 2 by sending to slave servers 1, 2, and 3 a request to communicate with the slave server having UIN bit value equal to UIN 22. Method 300 next moves to block 316, which shows slave system processors 1A and 3A receiving the request for slave server 2 and determining that UIN bit values for respective slave servers 1 and 3 are not equal to 22. Block 314 further shows slave system processors 1A and 3A directing modems 1B and 3B through I/O controllers 1C and 3C to change to the "receive" mode ("on-hook" mode) so that slave servers 1 and 3 are disconnected from

communication channel 150 and master system 100.

Following block 316, block 318 depicts slave system processor 2A receiving the request and determining that 5 UIN bit value for respective slave server 2 is equal to 22. Block 318 further illustrates modem 2B being maintained in the answer mode (off-hook mode) through I/O controller 2C so that slave server 2 stays connected with master system 100 through communication channel 150.

Method 300 proceeds to block 320, which shows master system 100 controlling slave server 2, for example, to 10 configure the performance monitoring of slave system processor, assign a particular task, etc. Method 300 finally ends at block 322.

A system, method, and program for establishing communication between data processing systems are disclosed. The present invention discloses in detail a system, method, and program for establishing modem communication between a master computer system and a plurality of slave computer systems coupled to a common serial communication channel. The present invention eliminates hardware and the need for a centralized communication management box with electronic or manual intervention and reduces expenses associated with 25 providing communication between a data processing system and a particular data processing system among a plurality of data processing systems.

While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that 30 various changes in form and detail may be made therein

5

10

without departing from the spirit and scope of the invention. For example, although aspects of the present invention have been described with respect to a computer system executing software that directs the functions of the present invention, it should be understood that present invention may alternatively be implemented as a program product for use with a data processing system. Programs defining the functions of the present invention can be delivered to a data processing system via a variety of signal-bearing media, which include, without limitation, non-rewritable storage media (e.g., CD-ROM), rewritable storage media (e.g., a floppy diskette or hard disk drive), and communication media, such as digital and analog networks. It should be understood, therefore, that such signal-bearing media, when carrying or encoding computer readable instructions that direct the functions of the present invention, represent alternative embodiments of the present invention.

卷之五